

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平 9 - 3 1 9 7 8 8

(43) 公開日 平成9年(1997)12月12日

(51) Int. Cl. 6	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F	17/50		G 0 6 F	15/60 6 5 4 G
G 0 3 F	1/08		G 0 3 F	1/08 A
G 0 6 F	15/16	3 7 0	G 0 6 F	15/16 3 7 0 N
H 0 1 L	21/027			15/60 6 0 4 Z
	21/82		H 0 1 L	21/30 5 4 1 J
	審査請求	未請求	請求項の数 2	O L (全 8 頁) 最終頁に続く

(21) 出願番号 特願平8-294907

(22) 出願日 平成8年(1996)11月7日

(31) 優先権主張番号 特願平8-76939

(32) 優先日 平8(1996)3月29日

(33) 優先権主張国 日本 (J P)

(71) 出願人 000190688

新光電気工業株式会社

長野県長野市大字栗田字舎利田711番地

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番1号

(71) 出願人 391042782

日本エヌ・ユー・エス株式会社

東京都港区海岸3丁目9番15号

(74) 代理人 弁理士 綿貫 隆夫 (外1名)

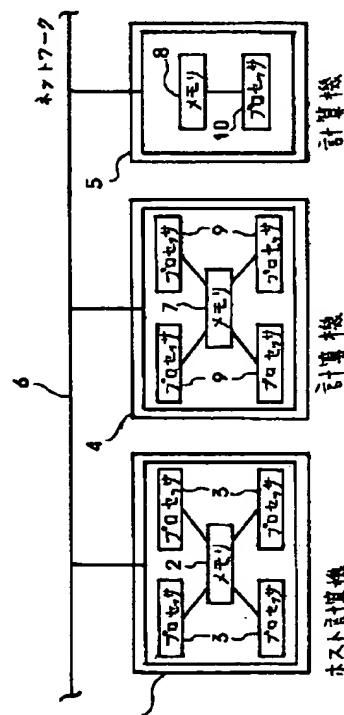
最終頁に続く

(54) 【発明の名称】 ネットワークによる並列処理システム

(57) 【要約】 (修正有)

【課題】 設計データのマスク製造データへの変換処理の高速化を図ったネットワークによる並列処理システム。

【解決手段】 主処理装置とネットワークで接続された複数の処理装置による並列処理により、設計データを加工してマスク製造データに変換処理する並列処理システムにおいて、前記ホスト計算機1は、設計データ領域を複数のブロックに分割すると共に、各ブロックの境界線を基準に外側に所定幅のマージンを設定して、該マージンの重なる領域にある図形データをその領域を共有する全てのブロックに当該図形データを処理対象として持たせ、メモリ7、8に記憶された各計算機4、5の各プロセッサ9、10の性能差や各ブロック内のデータ量を考慮して各ブロック毎のデータを各計算機4、5のプロセッサ9、10に振り分けて並列処理を行わせる。



【特許請求の範囲】

【請求項1】 主処理装置とネットワークで接続された複数の処理装置による並列処理により、設計データを加工してマスク製造データに変換処理するネットワークによる並列処理システムにおいて、前記主処理装置及びこれにネットワークで接続される各処理装置は、前記設計データを内部データにフォーマット変換して処理を行うプロセッサと、複数の設計パターン、処理プログラム、ネットワーク下の並列処理対象処理装置及びこれらの持つ各プロセッサに関するデータなどが記憶されたデータ記憶部と、を備え、前記主処理装置は、設計データ領域を複数のブロックに分割すると共に、各ブロックの境界線を基準に外側に所定幅のマージンを設定して、マージンの重なる領域にある図形データをその領域を共有する全てのブロックに当該図形データを処理対象として持たせ、前記データ記憶部に記憶された各処理装置の各プロセッサの性能差や各ブロック内のデータ量を考慮して各ブロック毎のデータを各処理装置のプロセッサに振り分けて並列処理を行わせることを特徴とするネットワークによる並列処理システム。

【請求項2】 前記ブロックの境界線より外側に設けたマージンの幅は、ブロック内の図形を太らせたり細らせたりする変位量の絶対値より大きい値に設定することを特徴とする請求項1記載のネットワークによる並列処理システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、主処理装置とネットワークで接続された複数の処理装置による並列処理により、LSI等の設計データを露光用のマスク製造データに変換処理するネットワークによる並列処理システムに関する。

【0002】

【従来の技術】従来より、例えばLSI設計用のデータを描画（露光）用のマスク製造データに変換処理し、該マスク製造データに基づき電子ビーム描画装置によりレチクル、マスク、ウェハなどに対して回路設計パターンを描画してLSIを製造する方法が用いられている。また、近年のLSIの大規模化（VLSI、ULSIの出現）、高密度化に伴い、LSI製造用の描画データ（露光データ）作成に対する処理時間の短縮化の要望が高まっている。このため、チップのデータ領域を多数のデータ処理対象領域にレイヤー毎に分割し、各処理対象領域を複数のプロセッサに割り当てて並列処理を行わせる技術、例えば、特開平2-232772号公報等に示す技術が提案されている。これは、LSIチップのデータ領

域を略同一のデータ量を有する小領域に分割し、並列プロセッサの負荷を一樣にして並列処理を行うものである。

【0003】

【発明が解決しようとする課題】しかしながら、上述した従来技術においては、以下に述べる課題がある。即ち、特開平2-232772号公報に示すLSIパターンデータの処理装置においては、並列プロセッサにより図形データのサイジング処理を行う際に、図形データの切断が生じたり、図形データの重なりが確認できないなどの不具合がある。また、LSIチップのレイアウトに応じてパターンデータ領域を略同一のデータ量を有する小領域に区分するための境界ラインの設定が複雑であり、効率の良い並列処理とはいえない。また、図形データ処理のため、同等の性能を有する並列プロセッサを装備しなければならず、設計コストも増大する。

【0004】本発明の目的は、上記従来技術の課題を解決し、設計データのマスク製造データへの変換処理の高速化を図ると共に、既存の設備を有効利用して設計コストの低減を図ったネットワークによる並列処理システムを提供することにある。

【0005】

【課題を解決するための手段】上記課題を解決するため、本発明は次の構成を備える。すなわち、主処理装置とネットワークで接続された複数の処理装置による並列処理により、設計データを加工してマスク製造データに変換処理するネットワークによる並列処理システムにおいて、前記主処理装置及びこれにネットワークで接続される各処理装置は、前記設計データを内部データにフォーマット変換して処理を行うプロセッサと、複数の設計パターン、処理プログラム、ネットワーク下の並列処理対象処理装置及びこれらの持つ各プロセッサに関するデータなどが記憶されたデータ記憶部と、を備え、前記主処理装置は、設計データ領域を複数のブロックに分割すると共に、各ブロックの境界線を基準に外側に所定幅のマージンを設定して、該マージンの重なる領域にある図形データをその領域を共有する全てのブロックに当該図形データを処理対象として持たせ、前記データ記憶部に記憶された各処理装置の各プロセッサの性能差や各ブロック内のデータ量を考慮して各ブロック毎のデータを各処理装置のプロセッサに振り分けて並列処理を行わせることを特徴とする。

【0006】また、前記主処理装置は前記ブロックの境界線より外側に設けたマージンの幅を、ブロック内の図形を太らせたり細らせたりする変位量の絶対値より大きい値に設定するのが好ましい。

【0007】上記構成によれば、主処理装置は、設計データ領域をレイヤー毎に複数のブロックに分割すると共に各ブロック内の多角形設計データを台形データの集合に分解し、各ブロックの境界線を基準に外側に所定幅の

マージンを設定して、該マージンの重なる領域にある図形データをその領域を共有する全てのブロックに当該図形データを処理対象として持たせて、データ記憶部に記憶された各処理装置の各プロセッサの性能差や各ブロック内のデータ量を考慮して各ブロック毎の図形データを各処理装置のプロセッサに振り分けて並列処理を行わせる。

【0008】

【発明の実施の形態】次に本発明に係るネットワークによる並列処理システムの一実施例について図面を参照して説明する。図1はネットワークによる並列処理システム構成例を示す説明図、図2は並列処理システム全体の処理の流れを示すフローチャート、図3はLSIチップ領域のブロック分割例を示す説明図、図4は図3から各ブロックを抜き出した説明図、図5は多角形図形データの台形データへの分解例を示す説明図、図6はブロック内の図形データのライブラリ参照例を示す説明図、図7～図10はマージンを持たないブロックとマージン付きブロックによる図形データのサイジング処理の結果を示す比較説明図である。

【0009】まず、図1を参照してネットワークによる並列処理システム構成例について説明する。本実施例はクラスタ構成の疎結合システムについて説明する。図1において、1は主処理装置としてのホスト計算機であり、複数の設計パターンをライブラリとしてファイルしたり、処理プログラムが記憶されたデータ記憶部としてのメモリ2、設計データを内部データにフォーマット変換して処理を行ったり、アプリケーションソフトを起動させたりする処理部としての複数のプロセッサ3を装備している。

【0010】4、5は上記ホスト計算機1とデータ通信回線6を介して接続された計算機であり、計算機4、5はデータ記憶部としてのメモリ7、8及びホスト計算機1からの指示された処理を行うプロセッサ9、10をそれぞれ装備している。これらの計算機4、5は、ネットワークにより接続された既存のワークステーション、パソコン等を処理装置として使用可能である。また計算機4、5は、メモリ容量が異なっていたり、プロセッサの数や処理性能等が異なっていたりしてもよい。これらの特性は、予めホスト計算機1のメモリ2に記憶されている。

【0011】なお、本システムのアプリケーションは、並列処理を実現するためのシステムの構成を特に制限するものではない。即ち、本実施例のように、処理装置としてシングルプロセッサ機やマルチプロセッサ機のうちいずれか或いは双方を使用してもよく、或いは1台のマルチプロセッサ機による密結合のシステムや超並列機等による構成であっても良い。また、システムに装備されるCPUの数は、サーバやハードによるもの以外は持たなくてもよく、既存の装置構成によりシステムを構築できる。

【0012】次に上記システムによる設計データよりマスク製造データの変換処理を行う動作の流れを図2に示すフローチャートを用いて説明する。LSIの設計パターンデータ(GDS)は、一旦システム独自の内部フォーマットデータにレイヤー単位に変換される。この内部フォーマットデータは、後述するように、LSIの設計領域をレイヤー毎に幾つかの小領域に分割し、分割された各小領域に図形データを分散して持つように構築されている。この小領域をブロックといい、1つの設計データを構築する複数のレイヤーは、全て共通のブロックに分割され、各ブロック単位でホスト計算機1より各計算機4、5等の各プロセッサに振り分けられる。上記ブロック内のデータは、ホスト計算機1より各計算機4、5等に必要なブロックだけ転送またはコピーされて、各ブロック単位で各プロセッサにより並列処理が行われる

(図1参照)。このとき、ホスト計算機1は、該ホスト計算機1とネットワークに接続された各計算機4、5間や各計算機4、5間のプロセッサ間に性能差があるとき、該プロセッサの性能差とブロックの図形データ数

(疎密)を考慮して、負荷の大きい(データ数の多い)ブロックを高性能のプロセッサへ、負荷の小さいブロックを比較的性能の低いプロセッサに自動的に振り分ける。プロセッサの数よりブロックの数の方が多いときは、1ブロックの処理を終えたプロセッサに次のブロックの処理が振り分けられ、全ブロックが変換処理されるまで以上の作業を繰り返す。これによって、並列処理の高速化を実現するものである。そして、内部フォーマットの複数のレイヤーはOR処理が施され、やはり内部フォーマットのマスク層が合成される。その際も処理はブロック単位で複数のプロセッサに振り分けて行われる。

【0013】また、ホスト計算機1及び各計算機4、5においては、上記フォーマット変換された内部フォーマットデータ(マスク単位)を入力して、ブロック内の図形データに対して図形間の論理処理、図形のサイジングが行われ、処理結果を内部フォーマットへ個々に出力する。また設計規則の検証も内部フォーマットデータに対して行われる。ここで、図形間の論理処理には、図形データに対してAND、OR、NOT等の処理を行うものが含まれ、図形のサイジングには図形を一定幅太らせたり細らせたりする処理が含まれ、設計規則の検証は、例えばサイジング等の処理を施した図形データに対して

「図形と図形との間が $0.0\mu\text{m}$ 以下の部分をチェックせよ」という個々のコマンドに応じて行われ、図形データが切断されたり、交差(重なり)が生じていないか、或いは鋭角のパターンが生じていないか等について検証を行い、警告を発したりして報知する。

【0014】本システムでは、ホスト計算機1及び各計算機4、5において、マスク単位の内部フォーマットデータに何らかの処理、例えば、図形間の論理処理、図形のサイジング、設計規則の検証等を行うとき、ブロック

内の図形データ処理のため他の計算機におけるブロックの処理状況や処理結果等の情報を全く必要としない。このように、ブロックに分割されたデータが互いに独立であるため、各計算機において効率の良い並列処理が可能となる。

【0015】次に上記ホスト計算機1や計算機4、5等において上記マスク単位の内部フォーマットデータに対する論理処理等が終わると、各プロセッサにより処理されたマスク単位の内部フォーマットデータは、ブロックどうしの図形間の重なり（具体的には後述するマージン領域で重複する図形）を除去すべく論理処理（OR処理）が施されて、マスク製造データが得られる。その際も、処理はブロック単位でホスト計算機1や計算機4、5等の各プロセッサに振り分けて行われる。このマスク製造データは、1つのファイル内に単位マスクデータをシーケンシャルに並べたもの（例えばMEBES、JEO L等）であっても、或いは複数のファイルによって構成されたものであっても良い。

【0016】次に、上記ホスト計算機1よりネットワークにより接続された計算機4、5等に並列処理を行わせるためのチップ領域のブロック分割方法について説明する。図3において、11はLSIのチップ領域を示すものであり、該チップ領域内に設計された回路パターンを示す多角形図形データを含む領域をブロック境界線12により任意のブロックに分割する。各ブロック内に存在する多角形図形データを幾つかの台形データに分解する方法を図5に示す。図5は多角形のうち凹部となるコーナー部p、qより水平線を引いて台形分解した場合を示す。また、内部フォーマットでは、各ブロックが直接的に持つ図形データ以外に、繰り返し参照できる図形パターンをライブラリに持つ。ライブラリは、1層のマスクに対してグローバルで各ブロックからライブラリ参照データを参照できる。このブロック内の図形データのライブラリ参照例を図6に示す。

【0017】また、上記ブロック境界線12により分割された各ブロックには、他のブロックとのブロック境界線12より外側に一定幅のマージン境界線13によるマージンを取り、隣接するブロックどうしブロック境界線12の内外にオーバーラップ領域14を形成させる。このとき各ブロックに形成されるマージンの幅は、図形のサイジング量の絶対値より大きい値に設定する。

【0018】上記図3に示すブロック境界線12により分割されたマージン付きブロック①～④を個々に抜き出した状態を図4に示す。ブロック分割の際、1つの多角形から分解された台形グループは、例えば多角形eのようにマージンによるオーバーラップ領域14を越えて複数ブロックにまたがって分割される場合がある。上記オーバーラップ領域にある台形データは、その領域を共有する全てのブロックに図形データとして持たせて処理が行われる。

【0019】ここで、上記チップ領域のブロック分割において、分割されたブロックにマージンを持たせた場合の効果について具体的に検証する。ブロック内の図形データに一例としてサイジング処理を施した場合、マージンを持たないブロックの場合とマージン付きブロックの場合の結果を比較して説明する。

【0020】図7において、(a)のようにブロック境界線12により上下2個のマージンのないブロックに分割された図形データA、Bをマイナスサイジング（一定幅細らせる）すると、(b)のように本来1つである図形データA、Bが切断された状態になる。これを避けるには、あるブロックの処理の際に隣接する全てのブロックのデータを意識しなければならず、ブロックを独立に扱い難くなる。

【0021】これに対し、図8において(a)に示すように、ブロック境界線12により上下2個のブロックに分割された各ブロックの外側にマージン境界線13によるマージンを形成した場合には、前述したように、マージンどうしのオーバーラップ領域にある台形データは、その領域を共有する全てのブロックに図形データとして持たせて処理が行われる。よって、上下2個のマージン付きブロックそれぞれに同様の図形データA、Bを持たせて処理が行われる。次に、(b)に示すようにマイナスサイジングを行った後、(c)に示すように上下ブロックどうしをブロック境界線12により接合すると、図形データA、Bを切断することなく図形データ全体のマイナスサイジングが行われる。なお、図形データA、Bを合成した結果、台形データは当初の2個から3個になる。従って、マージン付きブロックを用いると、ブロックを独立に扱っても不具合なく図形処理を行うことが可能になる。

【0022】次に、図9において、(a)のようにブロック境界線12により左右2個のマージンのないブロックに分割された図形データC、Dをプラスサイジング（一定幅太らせる）すると、(b)のように本来離れている図形データC、Dが交差（重なり合う）してしまった場合、検証のとき「図形が交差している」旨の警告を発する対象となる。しかしながら、ブロックを独立して扱う場合には、複数ブロック間でのそれぞれが持つ図形データ間の交差は確認できない。よって、あるブロックの検証を行うためには、隣接する全てのブロックのデータを参照する必要があるため、処理が複雑で時間がかかる。

【0023】これに対し、図10において(a)に示すように、ブロック境界線12により左右2個のブロックに分割された各ブロックの外側にマージン境界線13によるマージンを形成した場合には、前述したように、マージンどうしのオーバーラップ領域にある台形データは、その領域を共有する全てのブロックに図形データとして持たせて処理が行われる。よって、左右2個のマ

ジン付きブロックそれぞれに同様の図形データC、Dを持たせて処理が行われる。

【0024】次に、(b)に示すようにプラスサイジングを行うと、各ブロックにおいて、図形データC、Dの交差が生じたことが確認できる。従って、並列処理を行う各計算機において、設計規則の検証を行って適正な図形処理を行うことができるため、並列処理の高速化に寄与することが可能となる。

【0025】このように、チップ領域の図形データをブロック分割する際にマージン付きのブロックに分割して、図形データの並列処理を行うことで、処理の効率化を図り、並列処理の高速化に寄与することが可能となる。

【0026】以上、本発明の好適な実施例について種々述べてきたが、本発明は上述の実施例に限定されるのではなく、発明の精神を逸脱しない範囲で多くの改変を施し得るのももちろんである。

【0027】

【発明の効果】本発明は前述したように、主処理装置は、設計データ領域を複数のブロックに分割すると共に、データ記憶部に記憶された各処理装置の各プロセッサの性能差や各ブロック内のデータ量を考慮して各ブロック毎のデータを各処理装置のプロセッサに振り分けて並列処理を行わせる。従って、設計データのマスク製造データへの変換処理の高速化を図ると共に、既存の設備を有効利用して設計コストの低減を図ったネットワークによる並列処理システムを提供することができる。

【0028】また、チップ領域の図形データをブロック分割する際にマージン付きのブロックに分割して、図形データの並列処理を行うことで、処理の効率化を図り、並列処理の高速化に寄与することが可能となる。

【図面の簡単な説明】

【図1】ネットワークによる並列処理システム構成例を示す説明図である。

【図2】並列処理システム全体の処理の流れを示すフローチャートである。

【図3】LSIチップ領域のブロック分割例を示す説明図である。

【図4】図3のチップ領域から各ブロックを抜き出した説明図である。

10 【図5】多角形図形データの台形データへの分解例を示す説明図である。

【図6】ブロック内の図形データのライブラリ参照例を示す説明図である。

【図7】マージンを持たないブロックの図形データのマイナスサイジングの結果を示す説明図である。

【図8】マージン付きブロックの図形データのマイナスサイジングの結果を示す説明図である。

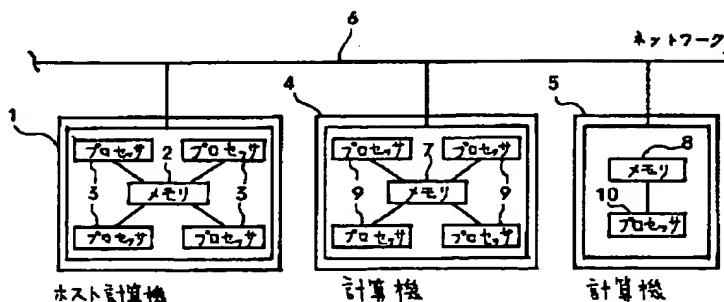
【図9】マージンを持たないブロックの図形データのプラスサイジングの結果を示す説明図である。

20 【図10】マージン付きブロックの図形データのプラスサイジングの結果を示す説明図である。

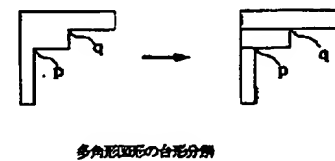
【符号の説明】

- 1 ホスト計算機
- 2, 7, 8 メモリ
- 3, 9, 10 プロセッサ
- 4, 5 計算機
- 6 データ通信回線
- 11 LSIのチップ領域
- 12 ブロック境界線
- 13 マージン境界線
- 14 オーバーラップ領域

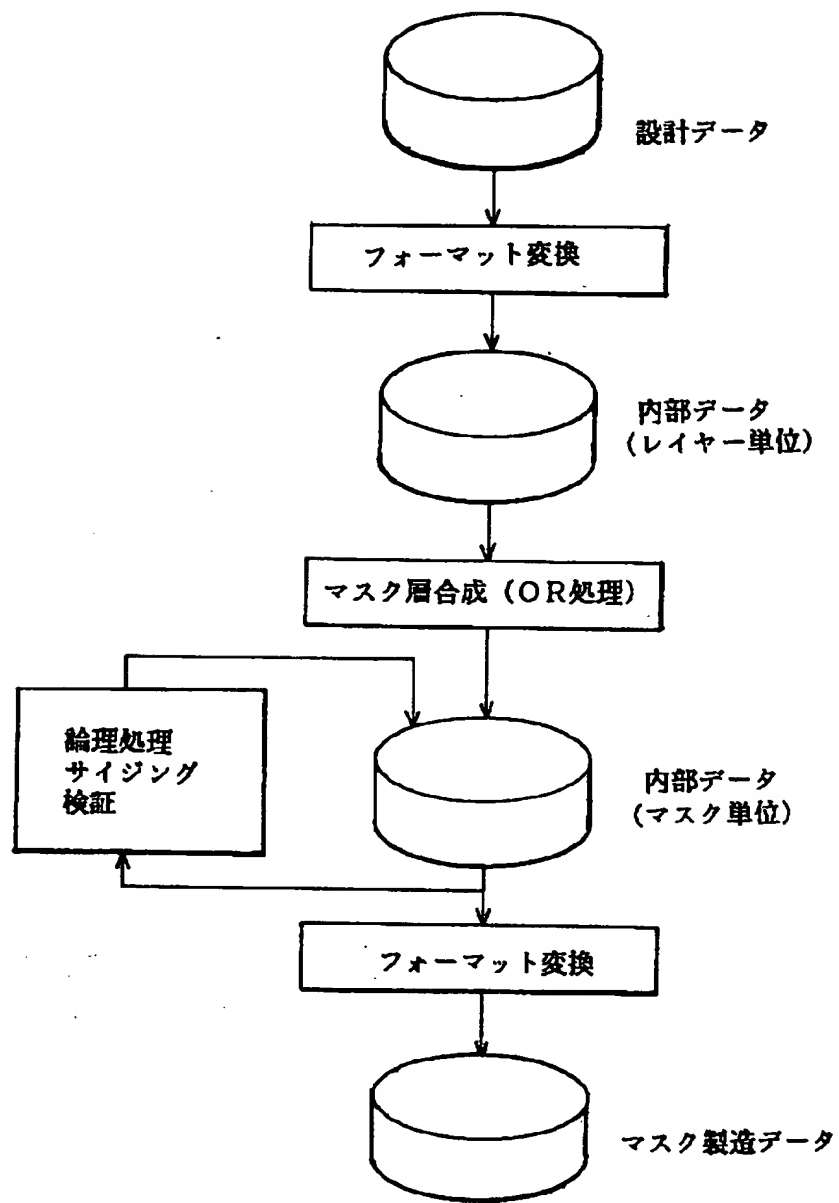
【図1】



【図5】

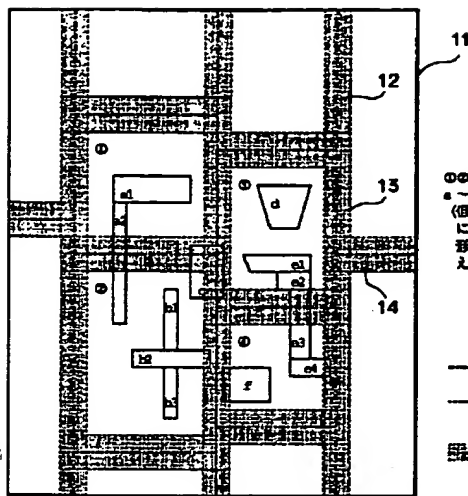


【図2】



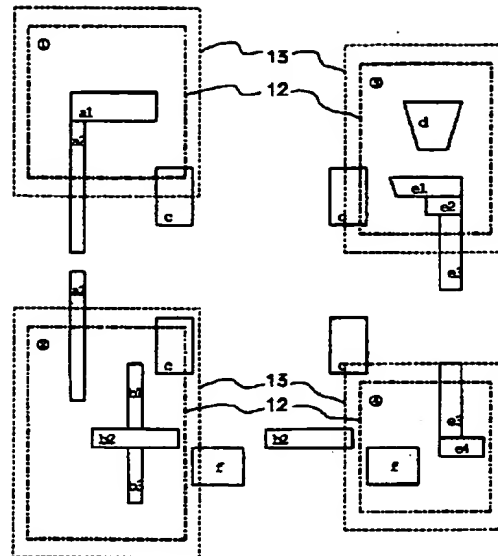
変換処理の流れ

【図3】

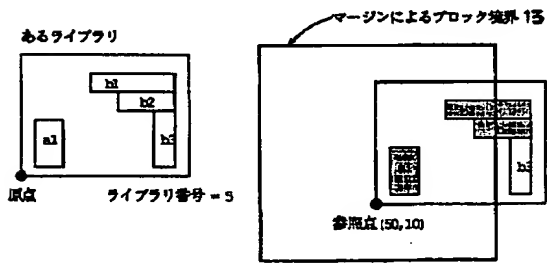


チップ領域のブロック分割例

【図4】

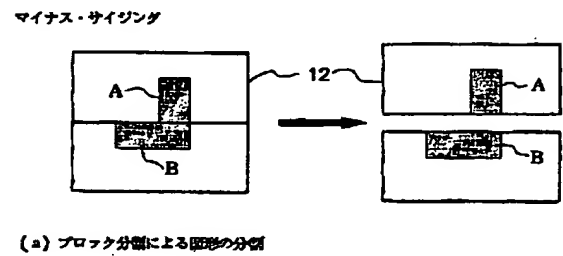


【図6】

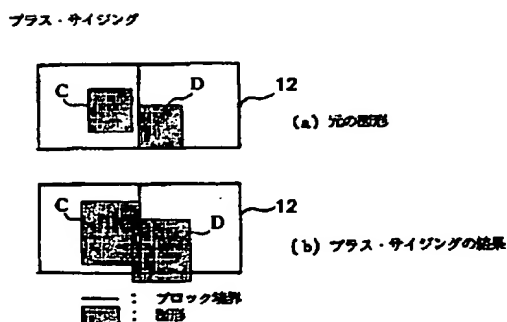


ライブラリとブロックからのライブラリ参照例

【図7】



【図9】

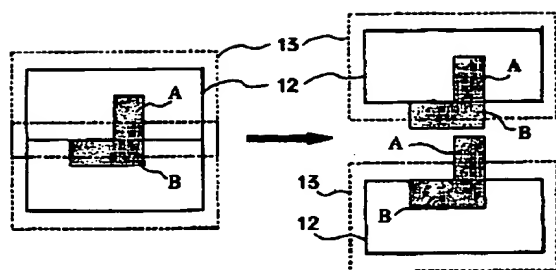


マージンのないブロック分割におけるプラス・サイジング

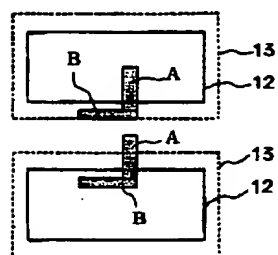
— : ブロック境界
 [] : 図形

マージンのないブロックによるブロック分割とマイナスサイジングの結果

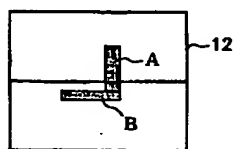
【図 8】



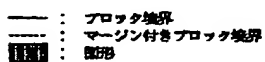
(a) マージン付きブロックによるブロック分割と図形の管理



(b) マイナス・サイジングの結果

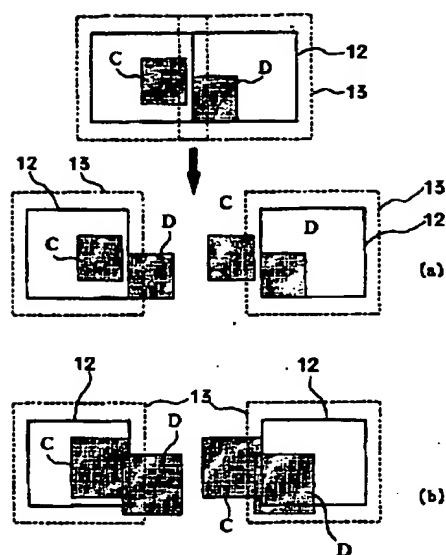


(c) 最終的な結果 (このとき台形は3個)

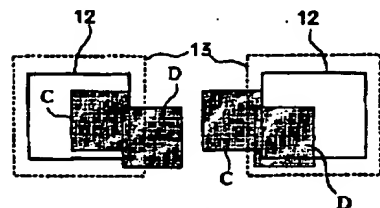


マージン付きブロックによるブロック分割とマイナスサイジングの結果

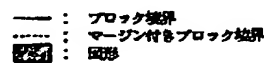
【図 10】



(a) マージン付きブロックによるブロック分割



(b) プラス・サイジングの結果



マージン付きブロックによるブロック分割とプラス・サイジングの結果

フロントページの続き

(51) Int. Cl. ⁶

識別記号

庁内整理番号

F I

技術表示箇所

H 0 1 L 21/30

5 4 1 M

21/82

C

(72) 発明者 望月 清貴

長野県長野市大字栗田字舎利田711番地
新光電気工業株式会社内

(72) 発明者 芥川 哲

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(72) 発明者 石原 靖文

東京都港区海岸3丁目9番15号 日本エ
ヌ・ユー・エス株式会社内